#8/Amdt B
T. McBeth-Brown
2/13/03

DOCKET: CU-1867 PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application: Eva KUHN]
]
Serial No: 09/269,485] GRP ART UNIT: 2151
]
Filed: March 29, 1999] Ex.: ZHEN, Li B.
]
For: COORDINATION SYSTEM]

SPECIFICATION-CLEAN VERSION

Please replace the Specification page 19, the second paragraph (lines 12-17) with the following paragraph.

> Activities are implicitly synchronized, because if a system depends on the result of another system, it knows which object contains the data, and it can simply perform an access to them. If the data are not yet available, a system wanting to access them will simply need more access time.

Please replace the Specification page 21, the first paragraph (lines 1-18) with the following paragraph.

> The communication objects can be of any type, i.e. they may be assigned a value only once or they may be updateable like variables. Each process can rely on the data read from the global space, because (in case of write-once objects) they will not change, or they are recoverable, respectively. Moreover, the communication objects have a unique identification number, but no global name. These object identification numbers (OID) are suitable exported via the above mentioned name servers, which are realized at application level by means of known database technologies. The communication objects are shared between processes by passing

them in arguments. A process having no reference to a communication object will not gain access to it. The agent maintaining the communication objects prevents processes from obtaining the reference to a communication object by trickery. This gives security, because only authorized processes are granted access to the data.

Please replace the Specification page 25, the third paragraph (lines 17-24) with the following paragraph.

B2

Regarding its general function, the global operating system 24 is shown in Fig. 3 and, in greater detail with regard to transaction control, in greater detail with regard to transaction control, in Figs. 4 to 9 in connection with Figs. 10 to 23 (transaction manager); the process manager is apparent from Fig. 24 in connection with Figs. 25 to 31, and the strategy manager (composed of single managers SMi for the corresponding distribution strategy) from Figs. 32 to 40.

Please replace the Specification page 29, the second paragraph (lines 17-27) with the following paragraph.

B3

In detail, initially at 44 it is asked whether the local request R is a request for the creation of a new object. If yes, the block 45 "object creation" (see Fig. 6) follows. If not, next it is asked whether the incoming local request is a request for "object read" (query 46). If yes, in block 47 (see Fig. 7) the command "read object" is executed. If not, as a third query it is checked at 48 whether the local request is a request for "alternative wait" (supervision of objects). If yes, the subprogram "alternative wait", block 49 (see Fig. 8), is called; if not, block 41 in Fig. 4 follows.

Please replace the Specification page 31, the third paragraph (lines 12-16) with the following paragraph.

It depends on the respective distribution strategy and its flags, whether, it is sufficient for the read request to check the local object structure, or whether communication steps have to be performed which ask other agents for the state and value of the object.

Please replace the Specification page 31, the fourth paragraph (page 31, line 17 to page 32, line 11 line 18) with the following paragraph.

In detail, with "object read" according to Fig. 7 it is initially tested at 50 whether the process is granted access to the object, and whether the object is of write-once type. If the result of this query is negative, at 51 an error message occurs, but if the result, is positive, at 52 a test takes place whether the object state is defined. If it is defined, at 53 the value of the object is returned, and control proceeds to the end of the function (block 41 in Fig. 5). If, however, the object state is not defined, at 54 it checked whether if the reading is blocking (i.e. whether the blocking flag is set), and if not, at 55 an error message occurs; if yes, in step 56 it is then tested whether the request has been issued by the user, which means that there exists no read request structure yet for this read request. If the result is negative, control proceeds to step 41; if the result is positive, according to block 57 a read request structure is created, which is then appended to the object. Then, according to block 58 the strategy manger is called to execute the function that the object is to be read. This step 58 will be illustrated in greater detail below, using Fig. 34.